Auto-Grouped Sparse Representation for Visual Analysis

Jiashi Feng[1], Xiaotong Yuan[2], Zilei Wang[3], Huan Xu[4], and Shuicheng Yan[1]

[1] Department of ECE, National University of Singapore
[2] Department of Statistics, Rutgers University
[3] Department of Automation, University of Science and Technology of China
[4] Department of ME, National University of Singapore

Abstract. In this work, we investigate how to automatically uncover the underlying group structure of a feature vector such that each group characterizes certain object-specific patterns, *e.g.*, visual pattern or motion trajectories from one object. By mining the group structure, we can effectively alleviate the mutual inference of multiple objects and improve the performance in various visual analysis tasks. To this end, we propose a novel auto-grouped sparse representation (ASR) method. ASR groups semantically correlated feature elements together through optimally fusing their multiple sparse representations. Due to the intractability of primal objective function, we also propose well-behaved convex relaxation and smooth approximation to guarantee obtaining a global optimal solution effectively. Finally, we apply ASR in two important visual analysis tasks: multi-label image classification and motion segmentation. Comprehensive experimental evaluations show that ASR is able to achieve superior performance compared with the state-of-the-arts on these two tasks.

1 Introduction

Most of current image analysis methods represent images by aggregating local features into image-level features, such as bag of words model [1–3]. These methods generally ignore the fact that the local features may be from different objects and treat the image-level feature as a whole in the follow-up computation. Such over-simplified strategy may render the results of image analysis inaccurate. For example, given two multi-object images containing one common object, they should be assigned one identical annotation. Indeed they are quite similar if only considering local features from the common object. But their image-level features may differ greatly due to involving local features from non-common objects and background and thus mislead the image similarity computation. To handle such mutual interference of multiple objects in image analysis, several previous works propose to perform segmentation or detection as pre-processing before feature extraction [4, 5]. However, such pre-processing is quite complicated, computationally expensive and inefficient.

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part I, LNCS 7572, pp. 640–653, 2012.

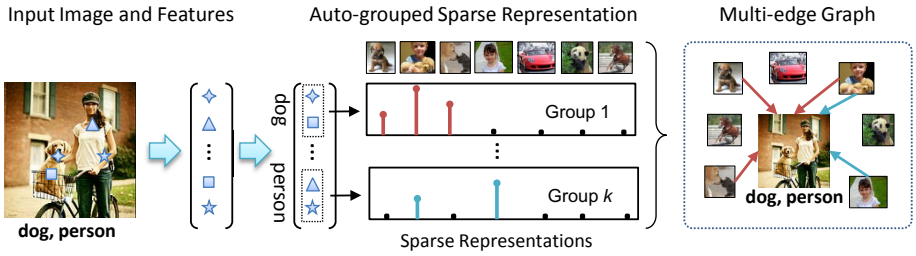


Fig. 1. Illustration on the proposed auto-grouped sparse representation method. The elements of the image-level feature represent different visual patterns. The feature elements are divided into k groups according to their individual sparse representations. Each group represents one specific object. Based on the group-wise sparse representations, a multi-edge graph is constructed to describe the relationship between the images.

In order to alleviate the mutual interference of multiple objects in given image-level features, we propose to divide its elements into several independent groups, such that each group represents typical visual patterns for one object. Thus, local features from different objects can be segregated to some extent and semantically different objects are considered independently in the follow-up computations. Then we can obtain analysis result (*e.g.*, image similarity) specific for one object, which is immune to the interference from other objects and background, as desired. In this way, we can obtain more accurate image relationship description from original image-level features and improve their performance in various visual analysis tasks.

To this end, we propose a novel auto-grouped sparse representation (ASR) method to *automatically learn* the intrinsic semantic groups of an image-level feature vector. The pursuit groups should roughly reside on the identical subspace if they correspond to objects from the same class [6–8]. And ASR computes multiple sparse representations for elements of an input image-level feature vector w.r.t. an over-complete basis to identify the subspaces (corresponding to the objects) [9]. In particular, ASR performs single sparse representation over each feature element, and meanwhile it imposes fusion-encouraging regularization to force the semantically correlated feature elements to share the same sparse representation. Thus the feature elements corresponding to the same object, namely falling on the same subspace, can be grouped together since they possess similar sparse representations.

Fig. 1 provides an example to illustrate the proposed ASR method. Given an input image, its feature elements representing the same object (*person* or *dog*) fall on identical subspace and thus will select identical basis images (containing *person* or *dog* respectively) in their sparse representations. Thus elements of its image-level feature can be divided into k groups according to their sparse representations. Each feature group includes several characteristic visual patterns for one specific object. In a multi-edge graph constructed based on the multiple sparse representation models, the input image is connected with the basis images

via varying number of edges, reflecting the relevance degree between them. It can be seen that such relationship is more accurate and flexible.

Note that the proposed ASR is a general method and can also be applied for other intrinsic group identification tasks, such as motion segmentation. In this work, we examine the applicability of ASR in two practical visual analysis tasks. The first application is to build the multi-edge graph [10] for more accurately classifying multi-label images. Compared with conventional single edge graph, our multi-edge graph achieves the state-of-the-art performance on the NUS-WIDE-LITE database. And the second one is two-view motion segmentation. ASR segments the motion trajectories by grouping the corresponding mixture linear regression models. Compared with previously well-performed methods [11, 12], ASR significantly decreases the segmentation error rates and offers more accurate and stable segmentation results.

2 Related Work

The proposed work aims at automatically uncovering the group structures across multiple feature entries and simultaneously calculating the underlying sparse representations within each group. The most intuitive approach to tackle this problem is the Expectation-Maximization (EM) method [11]. EM may regard the group assignments as hidden variables, and iterates the inference over hidden variables and the parameter estimation of decoupled models until a *local* optimum is reached. Gaffney *et al.* [13] applied the EM method to the trajectory clustering with the assumption that the motion trajectories are generated from a mixture regression model. The documentable limitation of the EM method is the locality of its optimization and thus the final solution is typically sensitive to initialization.

The second type of approaches may be based on convex relaxation. Recently, Quadrianto *et al.* [14] proposed to solve the regression model with mixture of several regression vectors by relaxing the assignment variables into continuous ones. Their experimental results show that the convex formulation performs better than the EM method on a number of benchmark datasets. However, their formulation seems hard to be generalized to sparse representation setting. Indeed, to the best of our knowledge, there has been no effort on solving the sparse mixture regression problem in a convex optimization framework.

Our method is directly inspired by the convex relaxation of clustering [15], where the authors employ the sparsity-inducing norms to enforce the fusion of data points. Sparsity-inducing norms have emerged as flexible tools that allow variable selection in penalized linear models [16, 17]. In this paper, we combine these lines of research into our framework of auto-grouped sparse representation.

3 Problem Formulation

In this work, we propose an auto-grouped sparse representation (ASR) method to automatically identify the intrinsic group structure of a given feature vector. In

particular, the elements of a feature vector $\mathbf{y} \in \mathbb{R}^p$ constitute K non-overlapped groups $\{\mathbf{y}_{\mathcal{C}_1}, \ldots, \mathbf{y}_{\mathcal{C}_K}\}$, each of which represents one specific object and admits a specific sparse representation $\boldsymbol{\omega}_k \in \mathbb{R}^n$ w.r.t. the sub-matrix of provided over-complete basis matrix $A \in \mathbb{R}^{p \times n}$. And $\mathcal{C}_k \subseteq \{1, \ldots, p\}$ is the feature element indices contained in the k^{th} group. We aim to find the groups of elements in $\mathbf{y}$ and simultaneously estimate their sparse representations by optimizing the following objective,

$$\min_{\{\mathcal{C}_k, \boldsymbol{\omega}_k\}_{k=1}^K} \left\{ \frac{1}{2} \sum_{k=1}^{K} \|\mathbf{y}_{\mathcal{C}_k} - A_{\mathcal{C}_k} \boldsymbol{\omega}_k\|_{\ell_2}^2 + \lambda \sum_{k=1}^{K} \|\boldsymbol{\omega}_k\|_{\ell_1} \right\}, \tag{1}$$

where $\mathbf{y}_{\mathcal{C}_k}$ denotes elements of $\mathbf{y}$ indexed by $\mathcal{C}_k$ and $A_{\mathcal{C}_k}$ denotes rows indexed by $\mathcal{C}_k$ in the matrix A. In the above optimization problem, each element of $\mathbf{y}$ is assigned to its corresponding group such that the overall loss is minimized.

The above objective function is a combinatorial optimization problem and in general computationally intractable. Following the relaxation technique introduced in [15], we relax the hard constraint on the number of groups to the fusion-encouraging constraint on the sparse representations $\{\mathbf{w}_i\}_{i=1}^p \subset \mathbb{R}^n$ of all elements in $\mathbf{y}$:[1]

$$\begin{aligned} &\min_{\{\mathbf{w}_i\}_{i=1}^p} \left\{ \frac{1}{2} \sum_{i=1}^{p} \|y_i - A_i \mathbf{w}_i\|_{\ell_2}^2 + \lambda \sum_{i=1}^{p} \|\mathbf{w}_i\|_{\ell_1} \right\}, \\ &\text{subject to}: \sum_{i<j} \mathbf{1}_{\mathbf{w}_i \neq \mathbf{w}_j} \leq t. \end{aligned} \tag{2}$$

Here y_i denotes the i^{th} element of the vector $\mathbf{y}$, A_i denotes the i^{th} row of the matrix A. The indicator function $\mathbf{1}_{\mathbf{w}_i \neq \mathbf{w}_j}$ takes value 1 if $\mathbf{w}_i$, $\mathbf{w}_j$ are unequal, and 0 otherwise; $\sum_{i<j}$ denotes $\sum_{i=1}^{p-1} \sum_{j=i+1}^{p}$. Intuitively, the constraint on the number of different vectors $\mathbf{w}_i$ serves as a proxy of constraining the number of groups. When $t \geq p(p-1)/2$, it amounts to each entry forming an individual group. Otherwise, along with the decrease of t, more feature elements are assigned into the same group.

However, due to the non-convexity of the indicator function, Problem (2) remains computationally hard. Here, we replace the indicator function by ℓ_∞-norm [15], which results in the following convex optimization problem:

$$\begin{aligned} &\min_{\{\mathbf{w}_i\}_{i=1}^p} \left\{ \frac{1}{2} \sum_{i=1}^{p} \|y_i - A_i \mathbf{w}_i\|_{\ell_2}^2 + \lambda \sum_{i=1}^{p} \|\mathbf{w}_i\|_{\ell_1} \right\}, \\ &\text{subject to}: \sum_{i<j} \|\mathbf{w}_i - \mathbf{w}_j\|_{\ell_\infty} \leq t. \end{aligned}$$

The constraint imposed by the ℓ_∞-norm encourages the maximal difference between two vectors to be zero, namely fusing them together. It can be equivalently expressed in following regularization form:

[1] More details and underlying rationale are referred to [15].

$$\min_{\{\mathbf{w}_i\}_{i=1}^p} \left\{ \frac{1}{2} \sum_{i=1}^{p} \|y_i - A_i \mathbf{w}_i\|_{\ell_2}^2 + \lambda \sum_{i=1}^{p} \|\mathbf{w}_i\|_{\ell_1} + \beta \sum_{i<j} \|\mathbf{w}_i - \mathbf{w}_j\|_{\ell_\infty} \right\}. \quad (3)$$

The objective function of Problem (3) consists of a smooth loss term and two non-smooth regularization terms. In particular, we decompose the objective function $f(\mathbf{w})$ into the following two terms:

$$\hat{f}(\mathbf{w}) := \frac{1}{2} \sum_{i=1}^{p} \|y_i - A_i \mathbf{w}_i\|_{\ell_2}^2,$$

$$r(\mathbf{w}) := \lambda \sum_{i=1}^{p} \|\mathbf{w}_i\|_{\ell_1} + \beta \sum_{i<j} \|\mathbf{w}_i - \mathbf{w}_j\|_{\ell_\infty}.$$

The problem bearing such non-smooth terms can be solved by smooth approximation [18]. We provide the optimization details in the following section. Using the proposed ASR, the feature element-wise sparse representations $\{\mathbf{w}_i\}_{i=1}^p$ are effectively recovered. In certain cases they may not exactly form distinct groups $\{\boldsymbol{\omega}_k\}_{k=1}^K$. However, it is still possible to construct reliable groups. In this work, we build an affinity graph of these representation vectors, and use a gap in the distribution of eigenvalues of the corresponding Laplacian matrix to estimate the number of groups K. Then spectral clustering techniques [19] can be applied to the affinity graph to cluster the representation vectors $\{\mathbf{w}_i\}_{i=1}^p$ into K groups. And we obtain the feature elements group $\{\mathbf{y}_k\}_{k=1}^K$ accordingly. Then $\{\boldsymbol{\omega}_k\}_{k=1}^K$ can be estimated by performing sparse representation on each group individually.

4 Optimization Procedure

4.1 Smooth Approximation

According to the smooth approximation proposed in [18], the non-smooth regularization term $r(\mathbf{w})$ can be approximated by the following smoothed one,

$$r_\mu(\mathbf{w}) = \lambda \sum_{i=1}^{p} s_\mu(\mathbf{w}_i) + \beta \sum_{i<j} q_\mu(\mathbf{w}_i, \mathbf{w}_j),$$

where

$$s_\mu(\mathbf{w}_i) := \max_{\|\mathbf{v}\|_{\ell_\infty} \le 1} \langle \mathbf{w}_i, \mathbf{v} \rangle - \frac{\mu}{2} \|\mathbf{v}\|_{\ell_2}^2, \quad (4)$$

$$q_\mu(\mathbf{w}_i, \mathbf{w}_j) := \max_{\|\mathbf{v}\|_{\ell_1} \le 1} \langle \mathbf{w}_i - \mathbf{w}_j, \mathbf{v} \rangle - \frac{\mu}{2} \|\mathbf{v}\|_{\ell_2}^2. \quad (5)$$

Herein, μ is a parameter to control the approximation accuracy and fixed as 1×10^{-4} throughout the experiments. For a fixed $\mathbf{w}_i$, denote $\mathbf{v}(\mathbf{w}_i)$ the unique maximizer of (4). It is standard that $\mathbf{v}(\mathbf{w}_i) = \min\{1, \max\{-1, \mathbf{w}_i/\mu\}\}$ where

operators $\max\{\cdot,\cdot\}$ and $\min\{\cdot,\cdot\}$ are performed in element-wise for the involved vectors. Moreover, $s_\mu(\mathbf{w}_i)$ is differentiable and its gradient $\nabla s_\mu = \mathbf{v}(\mathbf{w}_i)$ is Lipschitz continuous with the constant $L_s = 1/\mu$ [20]. Also, denote $\mathbf{v}(\mathbf{w}_i, \mathbf{w}_j)$ the unique maximizer of (5). Then $\mathbf{v}(\mathbf{w}_i, \mathbf{w}_j)$ can be easily obtained via the ℓ_1-ball projection algorithm [21]. Moreover, $q_\mu(\mathbf{w}_i)$ is differentiable and its gradient $\nabla q_\mu(\mathbf{w}_i) = \sum_{j \neq i} \mathbf{v}(\mathbf{w}_i, \mathbf{w}_j)$ is Lipschitz continuous with the constant $L_q = 1/\mu$ for each term [20].

4.2 Optimization of the Smoothed Objective Function

For a fixed small smoothness parameter μ, we are going to minimize the following smoothed objective function,

$$f_\mu(\mathbf{w}) := \hat{f}(\mathbf{w}) + r_\mu(\mathbf{w}). \tag{6}$$

It is known that $f_\mu(\mathbf{w})$ is differentiable with the gradient:

$$\nabla f_\mu(\mathbf{w}_i) = \nabla \hat{f}(\mathbf{w}_i) + \nabla r_\mu(\mathbf{w}_i), \tag{7}$$

where,

$$\nabla \hat{f}(\mathbf{w}_i) = A_i^T \left(A_i \mathbf{w}_i - y_i\right),$$
$$\nabla r_\mu(\mathbf{w}_i) = \mathbf{v}(\mathbf{w}_i) + \sum_{j \neq i} \mathbf{v}(\mathbf{w}_i, \mathbf{w}_j).$$

It is straightforward to verify that $\nabla \hat{f}(\mathbf{w})$ is Lipschitz continuous with constant $L_f = \|A^T A\|_2$, where $\|\cdot\|_2$ denotes the spectral norm of a matrix. Combining the discussion in the previous subsection, we get that $\nabla f_\mu(\mathbf{w}_i)$ is Lipschitz continuous with the constant,

$$L_{\hat{f}_\mu} = \|A^T A\|_2 + \frac{1}{\mu}\left(\lambda + \beta\right). \tag{8}$$

In particular, we employ the Accelerated Proximal Gradient (APG) method [22] to optimize $f_\mu(\mathbf{w})$. The detailed optimization procedure is provided in Algorithm 1.

4.3 Convergence Analysis

The following theorem guarantees the convergence of Algorithm 1.

Theorem 1. *Let the sequences $\{\mathbf{w}^{(t)}\}_{t=0}^{\infty}$ be generated by Algorithm 1. Then for any $t \geq 0$, we have,*

$$f_\mu(\mathbf{w}^{(t)}) - f_\mu(\mathbf{w}^*) \leq \frac{4 L_{f_\mu} \|\mathbf{w}^*\|_{\ell_2}^2}{(t+1)(t+2)},$$

where $\mathbf{w}^*$ is an optimal solution to the problem (6) and L_{f_μ} is the Lipschitz constant of the function $f_\mu(\cdot)$ calculated in Eqn. (8).

Algorithm 1. Smooth minimization for objective (3)

Input: $A \in \mathbb{R}^{p\times n}$, $\mathbf{y} \in \mathbb{R}^p$, λ, β, $\text{iter}_{\max}$ and ϵ.
Output: $\mathbf{w}_i, i = 1, \ldots, p$.
Initialization: Calculate L_{f_μ} according to Eqn. (8). Initialize $\mathbf{w}^{(0)} \in \mathbb{R}^p$, $\boldsymbol{\gamma}^{(0)} \in \mathbb{R}^p$, and let $\eta^{(0)} \leftarrow 1, t \leftarrow 0$.
repeat
$\boldsymbol{\alpha}^{(t)} = (1-\eta^{(t)})\mathbf{w}^{(t)} + \eta^{(t)}\boldsymbol{\gamma}^{(t)}$,
Calculate $\nabla f_\mu(\boldsymbol{\alpha}^{(t)})$ according to Eqn. (7),
$\boldsymbol{\gamma}^{(t+1)} = \boldsymbol{\gamma}^{(t)} - \frac{1}{\eta^{(t)} L_{f_\mu}} \nabla f_\mu(\boldsymbol{\alpha}^{(t)})$,
$\mathbf{w}^{(t+1)} = (1-\eta^{(t)})\mathbf{w}^{(t)} + \eta^{(t)}\boldsymbol{\gamma}^{(t+1)}$,
$\eta^{(t+1)} = \frac{2}{t+1}, t \leftarrow t+1$.
until $t > \text{iter}_{\max}$ or $|f_\mu(\mathbf{w}^{(t+1)}) - f_\mu(\mathbf{w}^{(t)})| < \epsilon$.

The above theorem can be directly derived from Theorem 2 in [18]. From Theorem 1, for a fixed μ, it can be seen that Algorithm 1 has the optimal rate of convergence $O(1/t^2)$, where t is the number of iterations. In terms of the desired residue, *i.e.*, $|f_\mu - \min f_\mu| \leq \epsilon$, the rate of convergence is $O(1/\sqrt{\epsilon})$.

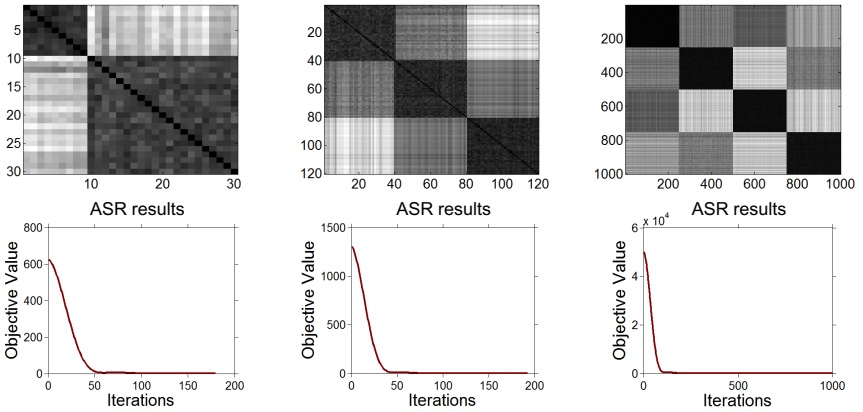
5 Experiments

5.1 Toy Problem: Sparse Mixture Regression

We first apply ASR in sparse mixture regression problem [14] to verify its effectiveness in uncovering data's group structure. The observed data points $\{y_i\}_{i=1}^N$ are generated according to the linear model $y_i = \boldsymbol{\omega}_k^T \mathbf{a}_i + \varepsilon$. Here $\mathbf{a}_i$ is a given regressor vector, $\boldsymbol{\omega}_k$ is selected from a mixture of sparse linear models $\{\boldsymbol{\omega}_k\}_{k=1}^K$ and $\varepsilon \sim \mathcal{N}(0,1)$ is added Gaussian noise. Here the data points $\{y_i\}_{i=1}^N$ can be stacked into a feature vector $\mathbf{y} = [y_1, \ldots, y_N]^T$ and the regressor vectors are stacked to form the basis matrix $A = [\mathbf{a}_1, \ldots, \mathbf{a}_n]^T$. The mixture regression aims to estimate the K regression models $\{\boldsymbol{\omega}_k\}_{k=1}^K$ according to $\{y_i, \mathbf{a}_i\}_{i=1}^N$. And simultaneously data points $\{y_i\}_{i=1}^N$ are separated into K groups in which the data points are generated by the same linear model. Namely, it aims to find the group structure of the input vector $\mathbf{y}$ according to the underlying linear regression models of its elements.

In this experiment, we apply ASR on the dataset generated by varying number of linear models with $K = 2$, 3, 4. The number of data points n is respectively set as 30, 120, 1000. Data dimension p is fixed as 10. Each element of $\mathbf{a}_i$ and $\boldsymbol{\omega}_k$ is i.i.d. sampled from a uniform distribution on the unit interval. The models $\{\boldsymbol{\omega}_k\}$ are sparsified by randomly zeroing half of their elements. The value of regularization parameters are set as $\beta = 1/p^2$ and $\lambda = 0.1$. And the convergence parameters are fixed as $\epsilon = 1 \times 10^{-4}$ and $\text{iter}_{\max} = 10,000$. Fig. 2 shows the curves of the objective function values in Eqn. (3) along the optimization iterations, and the obtained data groups. The clear block diagonal structure of the ℓ_∞ distance matrix of the uncovered linear models well demonstrates the ability

of ASR to cluster the mixed data correctly. From the convergence curve, it can be seen that objective function converges within less than 200 iterations, which shows satisfying convergence rate.



(a) $K = 2$ (b) $K = 3$ (c) $K = 4$

Fig. 2. Auto-grouped results from ASR on the synthetic datasets for sparse mixture regression. Top panel shows the ℓ_∞-distance matrices of the recovered regression models, where darker color means smaller distance. And bottom panel shows the convergence curves of the optimization processes.

5.2 Multi-edge Graph For Image Classification

Multi-edge Graph vs. Single-Edge Graph. A type of popular methods for image classification is to perform semi-supervised learning based on a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ [23, 24]. Here each vertex $v_i \in \mathcal{V}$ represents an image which is described by a feature vector $\mathbf{y}_i \in \mathbb{R}^p$. And the edge $e_{ij} \in \mathcal{E}$ from the i^{th} to j^{th} vertex, with weight w_{ij}, represents their similarity. In traditional graphs, such as k-NN graph and ℓ_1-graph [25], similarity of two vertices is calculated based on the feature-level measure and represented by a single edge. However, as pointed out in the introduction, multiple intrinsic groups may exist in one feature vector (corresponding to different objects or background), and more accurate similarity can be obtained based on group-wise measurement. Here, we propose to apply ASR to build a multi-edge graph $\hat{\mathcal{G}} = \{\mathcal{V}, \hat{\mathcal{E}}\}$ to more accurately and flexibly describe the relationship between images, and obtain better image classification performance.

In constructing the multi-edge graph, we apply ASR for each feature vector $\mathbf{y}_i$ of vertex v_i by treating the others as basis A. Then we obtain K representation vectors $\{\boldsymbol{\omega}_i^k\}_{k=1}^K$ and the corresponding element groups of $\mathbf{y}_i$. Here the j^{th} element in $\boldsymbol{\omega}_i^k$, $\boldsymbol{\omega}_i^k(j)$, represents the similarity between v_j and v_i w.r.t. the k^{th} feature group and we construct the edge e_{ij}^k according to $\boldsymbol{\omega}_i^k(j)$. Note that for different samples, the intrinsic group structure may be different. And thus the number of edges K between two vertices may vary.

After constructing the multi-edge graph, any graph regularized semi-supervised learning method can be employed to perform multi-label image classification [23, 24]. Since most of the methods operate on the graph adjacent matrix, which bears $2D$ structure, we also need to construct an adjacent matrix for multi-edge graph $\hat{\mathcal{G}}$ through properly selecting and merging the multiple edges. In practice, we first duplicate each edge e_{ij}^k into multiple edges, and the number of duplication is equivalent to the number of the feature elements in k^{th} group. In this way, any two nodes in $\hat{\mathcal{G}}$ are directly linked by identical number of edges. Then we adopt the Marginal Fisher Analysis (MFA) ratio [26] to evaluate the discriminative capability of the edges for multiple image class. After obtaining the edges ranking on the discriminative capability, we select the top $t = 100$ edges and combine them into single edge by summing their weights directly. The produced graph adjacent matrix is used for the semi-supervised learning on image classification.

Results. We compare the multi-edge graph with the k-NN and ℓ_1-graph [25] on the multi-label image classification task. The evaluations are performed on the public NUS-WIDE-LITE dataset [27], which consists of $55,615$ images and 81 different semantic labels. Here, we use $27,807$ images as labeled data and the remains are unlabeled as in [28]. The used 634-D feature is the concatenation of 225-D block-wise color moments (CM), 128-D wavelet texture (WT), 73-D edge direction histogram (ED), 64-D color histogram (CH) and 144-D color correlogram (CC). The k-NN graph is constructed by selecting $k = 3000$ nearest neighbors. For the ℓ_1-graph, the regularization parameter $\lambda = 0.1$ is selected from $[0.001, 10]$. In building the multi-edge graph, the parameters are fixed as $\lambda = 0.1$, $\beta = 2 \times 10^{-4}$, $\text{iter}_{\max} = 200$ and $\epsilon = 1 \times 10^{-4}$. In our experiments, it takes about 15 seconds to build the graph for one vertex on a PC with Quad CPU 2.83GHz and 8GB memory. An exemplar sub-graph of $\hat{\mathcal{G}}$ is shown in Fig. 3. In this sub-graph, several vertices are linked to the query vertex v_1 via 7 edges (the estimated number of groups $K = 7$), each of which measures the similarity between corresponding vertex and v_1 based on a certain feature group, as indicated in the legend. It can be seen from Fig. 3 that more semantically similar vertices (*e.g.,* v_2, v_4) have larger number of edges with larger weights to the vertex v_1. This is because these vertices (images) contain more similar objects to v_1, which is captured by ASR in constructing the multi-edge graph.

After duplicating the edges between two vertices into 634 edges (total dimension of adopted feature), the MFA ratio is calculated based on 20 positive and negative nearest neighbors of each vertex in $\hat{\mathcal{G}}$ [26], and top $t = 100$ edges are selected and combined. Then the popular Random Walk (RW) [24] and Entropic Graph Semi-Supervised Classification (EGSSC) [23] are used to perform semi-supervised learning on the multi-edge graph and baseline single-edge graphs. For EGSSC, the parameters are searched in the sets $\mu \in \{1 \times 10^{-8}, 1 \times 10^{-4}, 0.01, 0.1\}$ and $\nu \in \{1 \times 10^{-8}, 1 \times 10^{-6}, 1 \times 10^{-4}, 0.01, 0.1\}$ as in [23]. Classification performance is measured by the Mean Average Precision (MAP) [28] and shown in Table 1. It can be seen that the multi-edge graph significantly improves the multi-label image classification performance, for both two semi-supervised learning methods. In

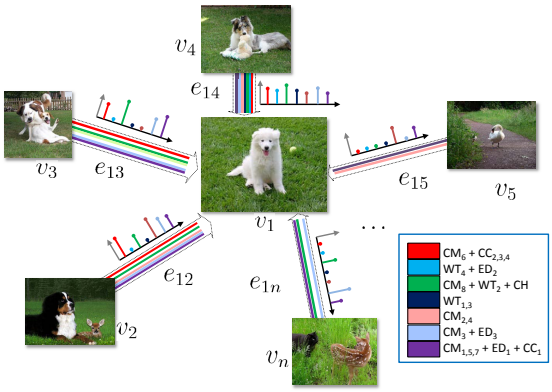


Fig. 3. A subgraph of the constructed multi-edge graph. Here 5 types of features are used. Note that for ease of display, each type of feature is shown in groups, as indicated by the subscripts in legend. The groups of these feature elements clusters obtained by ASR are shown in legend. In the multi-edge graph, the edges' weights are shown in a histogram form.

Table 1. MAP (%) of label propagation on different graphs

Graph	RW [24]	EGSSC [23]
kNN-graph	21.62	20.83
ℓ_1-graph	23.36	23.76
ℓ_1-graph_Comb	22.60	23.55
Multi-edge graph	**29.09**	**29.95**
LELR [28]	25.79	

particular, compared with the state-of-the-art performance from LELR [28], the improvement achieves 3.3% for multi-edge graph + RW and 4.1% for multi-edge graph + EGSSC.

Besides, we also compare ASR with k-means + ℓ_1-graph. In particular, the elements of feature vectors are clustered into 7 groups by k-means along the feature dimension. And we construct ℓ_1-graph for each feature element cluster. These ℓ_1-graphs are then combined into a 7-edge graph ℓ_1-graph_Comb for fairly comparing with our ASR multi-edge graph. From Table 1, it is shown that multi-edge graph outperforms ℓ_1-graph_Comb graph by about 6% MAP. This further demonstrates that ASR's ability to find reasonable feature groups with more discriminative information, benefitting from its accordance with the intrinsic structure of features.

5.3 Motion Segmentation

Two-View Motion Segmentation. Motion segmentation is aimed to assign multiple well tracked motion trajectories to the corresponding moving rigid

objects. From epipolar geometry, given two corresponding points $\mathbf{p}$ and $\mathbf{q}$ from two images $(\mathbf{p}, \mathbf{q} \in \mathbb{R}^3)$[2], they satisfy the following equation [29],

$$\mathbf{p}^T F \mathbf{q} = 0. \tag{9}$$

Here the fundamental matrix F encapsulates the intrinsic projective geometry between two views. Trajectories on the same object have identical fundamental matrix. And when K different rigid objects are moving independently, there are K different fundamental matrices $\{F_k\}_{k=1}^K$.

Here we apply ASR to the two view motion segmentation problem, where the tracked trajectories are only from two frames. We first rewrite the epipolar equation (9) for one corresponded pair as $(\mathbf{p} \otimes \mathbf{q})^T \boldsymbol{\omega} = 0$, where $\otimes$ denotes the Kronecker product and the vector $\boldsymbol{\omega}$ is formed by concatenating the columns of F. By removing the homogeneous coordinate (last element of $\boldsymbol{\omega}$) to the right hand side, the epipolar equation for N corresponding points can be written as $\mathbf{a}_i^T \boldsymbol{\omega}_k = 1$, where $\mathbf{a}_i$ consists of the first 8 elements of the vector $\mathbf{p}_i \otimes \mathbf{q}_i$. Here, we also denote the first 8 elements of original $\boldsymbol{\omega}$ as $\boldsymbol{\omega}$ without confusion. Then we stack the vectors $\mathbf{a}_i$'s into basis matrix $A = [\mathbf{a}_1, \ldots, \mathbf{a}_n]^T$ and the corresponding input vector is $\mathbf{y} = [1, \ldots, 1]^T \in \mathbb{R}^n$. Similar to the mixture regression, we can solve it through ASR as in Eqn. (1). Thus, we can obtain the segmentation of the motion trajectories according to their estimated fundamental matrix F_k, which is expressed as vector $\boldsymbol{\omega}_k$ in ASR.

Table 2. Segmentation errors (%) for sequences with 2 motions

Method	GPCA [12]	RANSAC	EM	ASR
Checkerboard: 78 sequences				
Mean	11.01	12.43 ± 0.26	37.44 ± 0.58	**9.07**
Median	7.51	8.22 ± 0.93	39.26 ± 0.82	**4.13**
Traffic: 31 sequences				
Mean	**7.75**	14.60 ± 1.12	41.24 ± 0.41	9.42
Median	**1.95**	10.54 ± 2.28	42.91 ± 0.52	2.32
Articulated: 11 sequences				
Mean	16.11	20.15 ± 0.61	33.77 ± 1.27	**6.15**
Median	14.14	17.28 ± 2.51	32.37 ± 4.08	**0.99**
All: 120 sequences				
Mean	10.63	13.70 ± 0.32	38.08 ± 0.42	**8.89**
Median	6.68	9.05 ± 0.98	40.33 ± 0.60	**3.07**

Results. We use the Hopkins155 dataset [30] to evaluate ASR for the two-view motion segmentation task. The dataset consists of 155 video sequences of two or three motions, which are divided into three categories: checkerboard, traffic, and articulated. We use the trajectories from the first 2 frames of each sequence as the input of the two-view motion segmentation.

[2] Actually, the point coordinates are in the projective plane, namely $\mathbf{p}, \mathbf{q} \in \mathbb{P}^2$.

We compare our method with three popular motion segmentation methods. The first one is Generalized Principal Component Analysis (GPCA) [12], which first projects the data points to a 4 dimensional subspace, and then groups the estimated normal vectors of the subspaces for data segmentation. The second method is the Expectation-Maximization (EM) [11], which is widely used but only provides a *local* optimum solution. The last one is the RANdom SAmple Consensus (RANSAC) which solves model fitting problem by random data sampling and evaluation [31]. In the experiments, both the EM and RANSAC are run 20 times and their average errors are reported. For ASR, the parameters are set as $\lambda = 0$ due to the fundamental matrix is not sparse, $\beta = 1/N^2$, $\text{iter}_{\max} = 1,000$ and $\epsilon = 1 \times 10^{-4}$. Note that here we do not compare the proposed method with multiple sample based methods, *e.g.*, sparse subspace clustering [32], since they only apply for multi-view motion segmentation. And they do not estimate the fundamental matrices since they vary across multiple frames.

The segmentation errors are provided in Table 2 for two motions and Table 3 for three motions respectively. It can be seen that ASR and GPCA significantly outperform the EM and RANSAC methods owning to their convexity. Compared with GPCA, the proposed ASR achieves smaller segmentation errors in most of the sequences, and brings 1.74% and 3.91% overall improvement for two and three motions respectively. More accurate segmentation results achieved by ASR well demonstrate its superior ability in uncovering the underlying data group structure.

Table 3. Segmentation errors (%) for sequences with 3 motions

Method	GPCA [12]	RANSAC	EM	ASR
Checkerboard: 26 sequences				
Mean	32.27	56.02 ± 0.29	46.88 ± 1.02	**25.53**
Median	30.92	57.47 ± 0.76	47.66 ± 1.85	**20.04**
Traffic: 7 sequences				
Mean	**17.58**	48.61 ± 1.24	47.56 ± 2.19	26.48
Median	**18.54**	51.37 ± 0.98	51.31 ± 1.32	29.92
Articulated: 2 sequences				
Mean	26.14	61.70 ± 3.83	43.27 ± 5.60	**10.05**
Median	26.14	61.70 ± 3.83	43.27 ± 5.60	**10.05**
All: 35 sequences				
Mean	28.86	54.62 ± 0.32	46.81 ± 1.11	**24.83**
Median	24.32	57.07 ± 0.79	48.28 ± 1.84	**22.62**

6 Conclusions and Future Work

In this work, we proposed auto-grouped sparse representation (ASR) to automatically obtain the underlying group structures of the correlated feature elements. In ASR, each uncovered group represents a certain semantically meaningful pattern. We applied a convex relaxation to the primal intractable objective function

to guarantee a global solution and further introduced smooth approximations to ease the optimization process. Furthermore, two realistic applications of ASR were considered besides the evaluations on synthetic data. For multi-label image classification, ASR achieves remarkable performance improvement over the state-of-the-art methods owing to its ability to more accurately describe the semantic relationship between images by building informative multi-edge graph. And for two view motion segmentation, ASR significantly reduces segmentation errors compared with previous methods. Our proposed ASR need include a set of pair-wise regularizations which may be inefficient for large-scale problems. In the future, we plan to explore how to enhance the efficiency of the ASR, *e.g.*, by utilizing some priors to remove redundant constraints. And we are also interested in providing the solution path for a general instruction on the parameter selection of λ and β.

Acknowledgments. This research is partially supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office. H. Xu is partially supported by National University of Singapore startup grant R-265-000-384-133.

References

1. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR (2005)
2. Perronnin, F., Sánchez, J., Mensink, T.: Improving the Fisher Kernel for Large-Scale Image Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
3. Yang, J., Yu, K., Huang, T.: Efficient Highly Over-Complete Sparse Coding Using a Mixture Model. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 113–126. Springer, Heidelberg (2010)
4. Li, F., Carreira, J., Sminchisescu, C.: Object recognition as ranking holistic figure-ground hypotheses. In: CVPR (2010)
5. Song, Z., Chen, Q., Huang, Z., Hua, Y., Yan, S.: Contextualizing object detection and classification. In: CVPR (2011)
6. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. TPAMI (2008)
7. Yuan, X., Yan, S.: Visual classification with multi-task joint sparse representation. In: CVPR (2010)
8. Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T., Yan, S.: Sparse representation for computer vision and pattern recognition. IEEE (2010)
9. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: CVPR (2009)
10. Liu, D., Yan, S., Rui, Y., Zhang, H.: Unified tag analysis with multi-edge graph. In: MM (2010)
11. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. J Roy. Stat. Soc. B Met. (1977)
12. Vidal, R., Ma, Y., Sastry, S.: Generalized principal component analysis (gpca). TPAMI (2005)

13. Gaffney, S., Smyth, P.: Trajectory clustering with mixtures of regression models. In: KDD (1999)
14. Quadrianto, N., Caetano, T., Lim, J., Schuurmans, D.: Convex relaxation of mixture regression with efficient algorithms. In: NIPS (2009)
15. Hocking, T., Vert, J., Bach, F., Joulin, A.: Clusterpath: an algorithm for clustering using convex fusion penalties. In: ICML (2011)
16. Shen, X., Huang, H.: Grouping pursuit through a regularization solution surface. J Am. Stat. Assoc. (2010)
17. Vert, J., Bleakley, K.: Fast detection of multiple change-points shared by many signals using group lars. In: NIPS (2010)
18. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. (2005)
19. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: NIPS (2001)
20. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge Univ. Pr. (2004)
21. Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the ℓ_1-ball for learning in high dimensions. In: ICML (2008)
22. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. Submitted to SIAM J. Optimiz. (2008)
23. Subramanya, A., Bilmes, J.: Entropic graph regularization in non-parametric semi-supervised classification. In: NIPS (2009)
24. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Tech. Rep. (2002)
25. Yan, S., Wang, H.: Semi-supervised learning by sparse representation. In: SDM (2009)
26. Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. TPAMI (2007)
27. Chua, T., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of singapore. In: CIVR (2009)
28. Chen, X., Yuan, X., Chen, Q., Yan, S., Chua, T.: Multi-label visual classification with label exclusive context. In: ICCV (2011)
29. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge Univ. Press (2000)
30. Tron, R., Vidal, R.: A benchmark for the comparison of 3-d motion segmentation algorithms. In: CVPR (2007)
31. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM (1981)
32. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: CVPR (2009)